05｜音频降噪如何对症下药？

2021-12-01 冯建元

《搞定音频技术》 课程介绍 >

讲述：冯建元
时长 16:48　大小 15.40M

你好，我是建元。今天我们来聊聊噪声的分类与常见的降噪方法。

听到噪声，很多人可能首先想到的是深夜的广场舞曲、呼啸的东北风、车水马龙的呼啸而过。但其实噪声是一个相对的概念。如果你想听的目标只有人们说话的声音，那么所有其它的声音包括音乐、 风声等其就都是噪声，而如果你想听鸟语虫鸣，那人声对你来说也是噪声。

在这里我们主要聊的是当保留目标是人声时，噪声会有哪些分类，它们有什么特点，以及我们如何更好地保留人声去除噪声。这也是音视频工程师的主要工作场景。

噪声的分类

从通信系统的角度来说，噪声可以分为**加性噪声和乘性噪声**。

加性噪声与信号之间满足加性条件，即**加噪信号是由噪声和源信号相加得到的**，这种情况下**信号和噪声是不相关的**，我们常见的自然噪声、人造的噪声如电子元器件发出的热噪声等都是这种。

乘性噪声则是，噪声和信号是相关联的，比如信号的衰减、房间的混响、多普勒效应等。这类噪声往往是以信号乘积的形式出现，而且往往是从信道传输中产生，所以也叫信道噪声。**我们这里主要讲的是人声与其它不相关的噪声的处理，所以主要是针对加性噪声来讲**。

加性噪声的种类如果按照声源，比如风声、汽笛声、键盘敲击声等，种类则成千上万，但从降噪方法的选择角度上来说，我们可以按照噪声是否平稳，把噪声分为两类：**稳态噪声和非稳态噪声**。

> 稳态噪声：比如手机、电脑之类的设备底噪、电脑散热器的风扇声等等。**它们一直存在且基本上响度、频率分布等声学特性都不随时间变化或者变换缓慢**。
>
> 非稳态噪声：比如开关门的声音、背景的人声、门铃声等等。**这些噪声的统计特性随时间而变化**。就好像你在一家餐馆里很多人在说话，还时不时掺杂着吃饭的时候碗碟碰撞的一些声音。
> 非稳态噪声按照是否连续又可以再分为**连续性非稳态噪声和瞬态噪声**，比如持续性的背景人声就是连续的噪声，而一些敲击声只会出现极其短暂的时间则为瞬态噪声。

稳态噪声和非稳态噪声的时域图和频域图如图 1 所示：

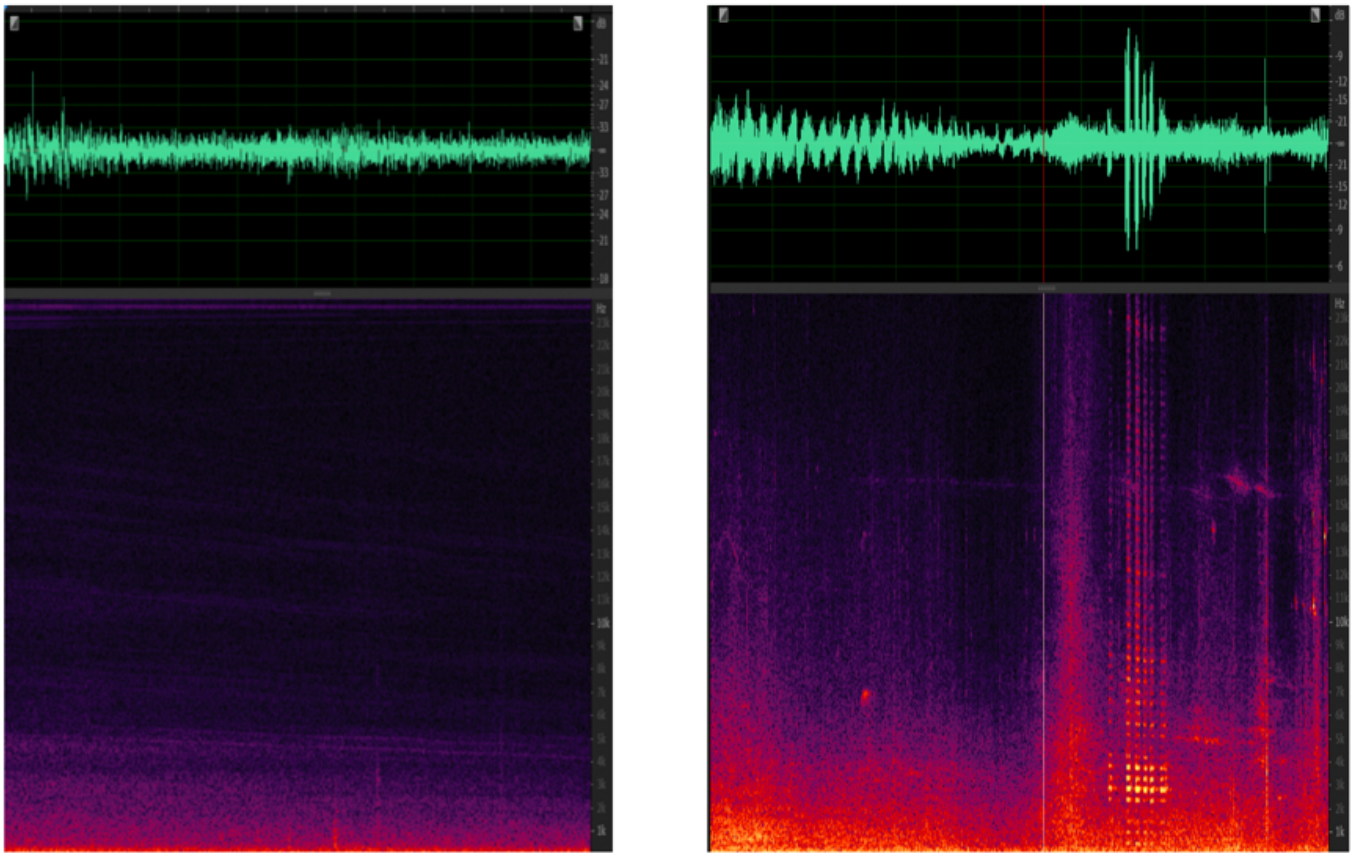

图1 稳态噪声（左），非稳态噪声（右）

很显然，稳态噪声由于在时间维度上没有变化，我们很容易通过对之前出现过的噪声进行建模，然后用相同的模型来对以后出现的噪声来进行抑制。而非稳态噪声尤其是瞬态噪声，则需要更多的依靠来区分其和正常语音之间的差异。如果更像是语音，则将其保留，反之则将其抑制。

值得注意的是，这些噪声往往不是单独存在的，可能你的手机底噪是一直存在的，同时你又在人声鼎沸的地铁站，那这时候稳态和瞬态噪声就都会存在。

如何降噪

清楚了噪声的分类后，接下来让我们看看目前有哪些常见的降噪算法，以及它们在降噪的能力上有哪些不同。这里我们重点介绍基于统计模型的实时降噪算法，这是因为在实时音频降噪处理时主要应用的就是这种算法。另外，基于机器学习的降噪是基于 AI 的，因此之后会单独开一讲。至于其他三种算法，这里你有个简单了解就足够了。

降噪算法第一招：线性滤波器

这在一些音频采集硬件的噪声处理中十分常见，因为硬件厂商知道自己的硬件噪声特性。比如由于电路设计在一些频段会有持续的电流声，这时可以采用一些比如高通滤波器来消

除低频噪声、用一些陷波滤波器来消除某些频段的持续噪声。

线性滤波器的处理方法算力要求十分低，但必须事先知道噪声会在哪个频段出现。所以在实际使用中一般会先做噪声频段检测，看看噪声出现在哪个或哪些频段，再设计线性滤波器或滤波器组来消除噪声。

降噪算法第二招：谱减法

谱减法的核心思想是先取一段非人声段音频，记录下噪声的频谱能量，然后从所有的音频频谱中减去这个噪声频谱能量。这种方法对稳态噪声比较有效果。但如果是非稳态噪声就会导致有的地方频谱减少了噪声有残留，有的地方频谱减多了人声有损伤。所以**谱减法一般用于离线稳态噪声的降噪处理。**离线的时候可以人工对音频进行分片处理，在每一个分片中噪声可以控制成稳态的。而在实时音频处理的时候，噪声状态经常是随时间变化的，我们很难让噪声一直保持绝对稳态。

降噪算法第三招：基于统计模型的实时降噪算法

这类算法是实时音频降噪时最常用的算法类别。算法的思想就是利用统计的方法估算出音频频谱中每个频点所对应的噪声和语音的分量。基于统计的降噪方法其实都是针对相对平稳的噪声进行去除，且为了方便找出噪声和人声的直观统计区别，一般都需要基于两个假设。

第一个假设：噪声相对于人声一定是在时域和频域上的声学统计特性都更平稳。

第二个假设: 所有的噪声都满足加性条件。

所以基于这两个假设，我们就可以解释很多我们平时在使用这些降噪算法时所遇到的现象。比如噪声中的瞬态噪声很难被抑制，比如敲桌子的声音、键盘声之类的。再比如在一些混响比较大的房间，听不出混响，且人声的失真也比较严重。了解了以上特性之后，让我们看看常见的几种基于统计的降噪。

这里我主要介绍一下常用的分位数噪声估计和维纳滤波（Quantile Noise Estimation and Winner Filter）以及一些改进方法，比如 OMLSA & IMCRA（Optimally Modified Log-Spectral Amplitude Estimator and Improved Minima Controlled Recursive Averaging）。分位数噪声估计和维纳滤波这种方法是 WebRTC 中自带的降噪算法。

维纳滤波是根据最小均方误差 MMSE 准则（滤波器的输出信号与需要信号之差的均方值最小）设计的线性滤波器。我们这里用到的是实时频域维纳滤波器，目标就是求出当前帧每个频点的能量有多少占比是语音，即语音的先验信噪比（SNR）。我们把当前帧的含噪信号与噪声的信噪比叫做后验信噪比，而纯净语音信号与噪声的信噪比叫先验信噪比。

因为在实时处理时没有纯净的参考信号，所以先验信噪比通常为后验信噪比结合判决引导的方法来估算。那么根据维纳滤波的原理降噪的步骤也就变成了从动态平滑的噪声模型得到噪声信号，然后根据含噪信号和噪声模型经过维纳滤波器进行降噪。

WebRTC 中的降噪流程如图 2 所示。

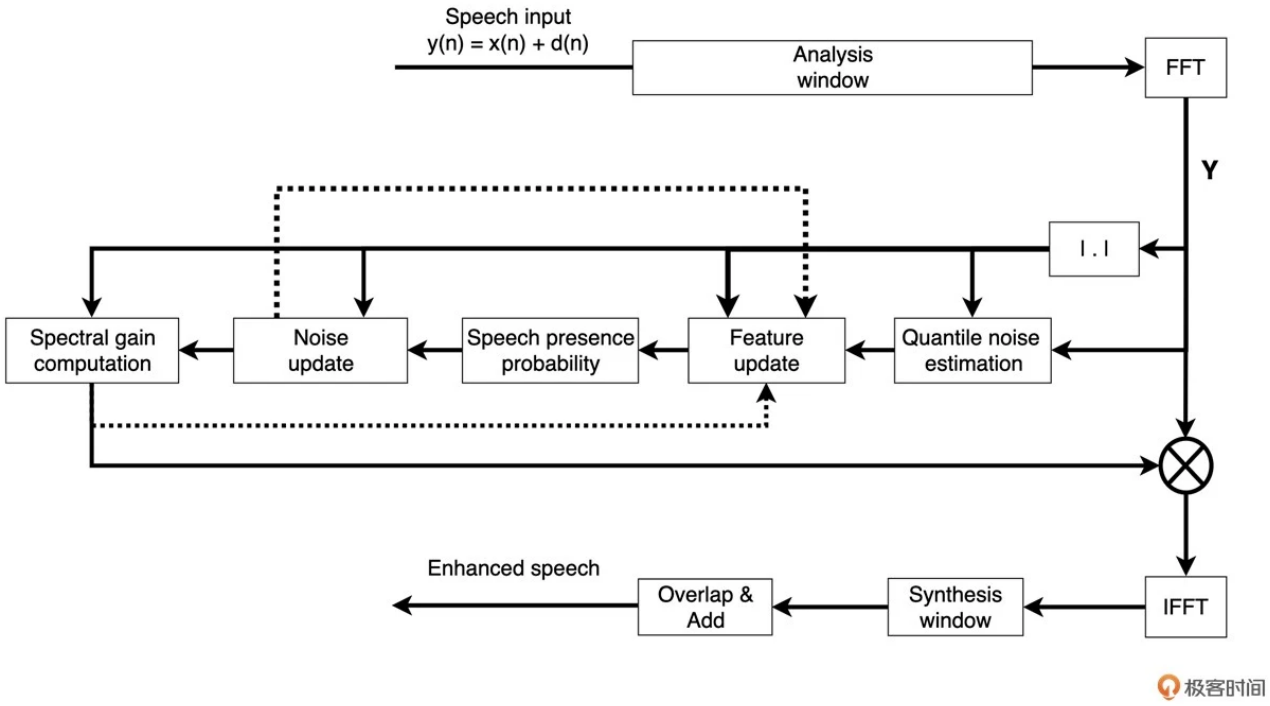

图2 WebRTC 中分位数噪声估计和维纳滤波的降噪算法处理流程

首先，做短时傅里叶变换（STFT），即对带噪信号加窗。接着，做快速傅里叶变换（FFT）再求模得到带噪信号的功率谱。然后，利用功率谱进行分位数噪声估计 (Quantile Noise Estimation)、语音存在概率（Speech Presence Probability）估计、噪声更新（Noise Update）以及噪声抑制系数计算（Spectral Gain Computation）。最后，把得到的每个频点的抑制系数乘以带噪信号的频率谱得到降噪后的频率谱。再做逆短时傅里叶变换（ISTFT）即可得到降噪后的时域信号。

这里就不一一展开了，内容很多，具体每一步怎么做你可以参考 Google 的WIPO 专利（Noise Suppression Method and Apparatus Using Multiple Feature Modeling for

Speech/noise Likelihood），里面和 WebRTC 开源库中的代码基本可以一一对应。

我们这里主要讲一下使用分位数噪声估计和维纳滤波的降噪算法逻辑背后的思考。

基于统计的降噪最主要的是对噪声进行实时建模，这个建模基于假设一，也就是说只对稳态的噪声进行建模。噪声模型迭代不能太快，比如这里 WebRTC 所用的分位数噪声估计都是在时频域上进行更新，且其更新周期大概为 700ms 左右。从听感上来说，如果噪声发生了变化，比如突然变大了，模型可能需要约 500ms～4s 来收敛到新的噪声模型，在这期间我们可能会听到一些噪声的残余。

这里降噪的理念和我们之前讲的谱减法有些类似，就是利用无人声段进行噪声的估计。这样在实时处理中就需要在无人声段进行噪声模型的迭代。即更新噪声模型以适应非稳态噪声的时变性。

那么为了区分人声和非人声就需要做一个人声判别也就是我们常说的 VAD（Voice Activity Detection）。这里的 VAD 就是利用几个人工提取的特征来进行统计得出的语音存在概率来判断的。这几个特征包括：频谱平坦度（Spectral Flatness ），频谱差异度（Spectral Difference），以及根据先验、后验信噪比的差异得出的似然因子 LR（Likelihood Ratio）Factor。具体如图 3 所示：

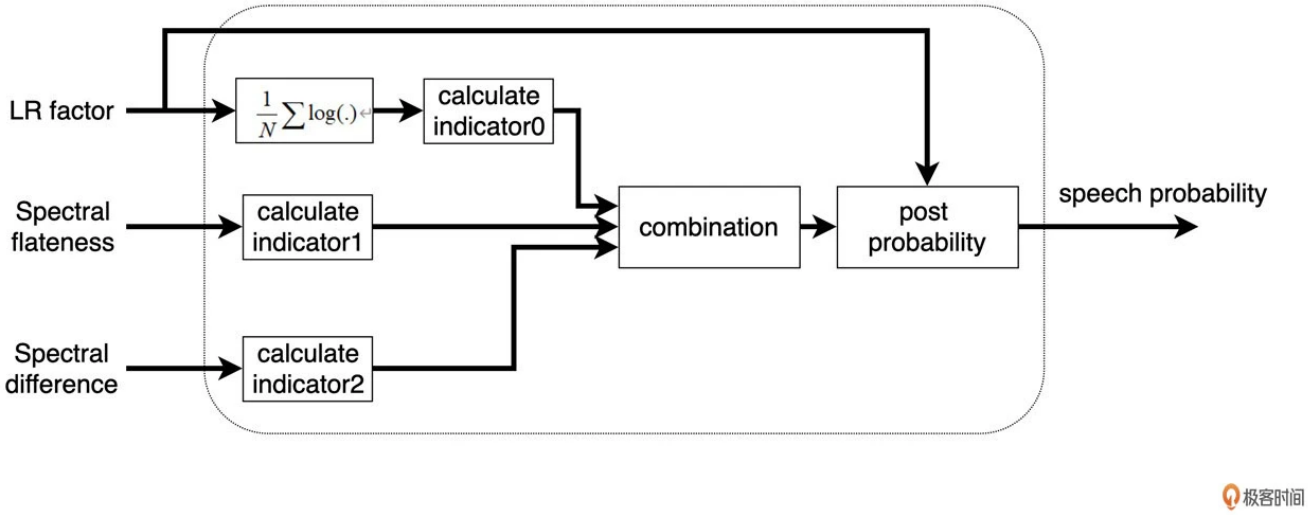

图3 Speech Probability的计算流程

似然因子在频域计算 log 均值得到 indicator 0，而根据频谱平坦度和差异度则可以分别得到 indicator 1 和 2。语音概率值就是根据这几个指标的加权平均（对应图中的

combination）和当前帧的似然因子来更新的。在实际计算中，当这个概率比较大时，也就是语音存在的可能性比较大，噪声模型的更新就很缓慢，反之则更新速度比较快。

噪声模型则是通过分位数噪声估计来得到。“Quantile”是英文中 1/4 的意思。这里也就是通过一个经验假设在噪声能量谱中，里面能量最小的 1/4 是稳态噪声。我们应该用这个部分来迭代更新初始噪声模型。

利用初始噪声模型以及含噪语音就可以得到频谱平坦度、频谱差异度，以及对数似然比特征，进而得到语音存在概率。有了语音存在概率，就可以更新噪声模型。有了噪声模型和含噪信号，那么根据加性假设，干净的语谱则是含噪信号减去噪声信号或者说含噪频谱乘以频谱增益。

了解了算法原理后，我们基本上就可以总结出 WebRTC 原生降噪算法的 3 个特点了：

1. 由于 speech probability 的判断降噪在有人声的地方基本不会进行噪声模型的更新，从而不会对语音造成损伤也就是说基本不会吃字，但是如果噪声是在说话的时候发生了变化，那么噪声无法被有效消除。
2. 基于 MMSE 的维纳滤波器有一个弊端，那就是对于浊音谐波间的噪声可能会有残留。这也就是为什么如果观测频谱的时候在谐波之间会有噪声的能量残留，这种噪声残留会随语音出现，听上去像是给语音加了伴奏。我们通常也把这种残留叫做音乐残留。
3. 由于在低信噪比的时候，语音存在概率的判断会失效，那么就会产生比较大的语谱损伤。
   现在你在使用 WebRTC 的时候遇到噪声残留和音乐噪声大致就能判断出原因了。那么有没有什么更好一点的方法可以把 WebRTC 的原生降噪改造一下呢？

这里简单介绍一下改进方法 OMLSA&IMCRA 算法。

它是由 Israel Cohen 提出的音频降噪算法。OMLSA 是对人声进行估计，通过先验无声概率及先验信噪比 SNR 的估计来得到有声条件概率，从而实现了对人声谱的估计。IMCRA 则是通过信号的最小值跟踪，得到先验无声概率估计和先验信噪比估计，来计算得到条件有声概率，进而获取噪声谱的估计。

将 OMLSA 同 IMCRA 相结合最后相当于是功率谱中最小点的追踪。这样藏在谐波之间的音乐噪声的能量由于明显小于谐波的能量就可以被去除了，这也是为什么这一算法可以有

效减少音乐噪声。

关于 WebRTC 原生降噪和 OMLSA&IMCRA 降噪的对比如图 4 所示：

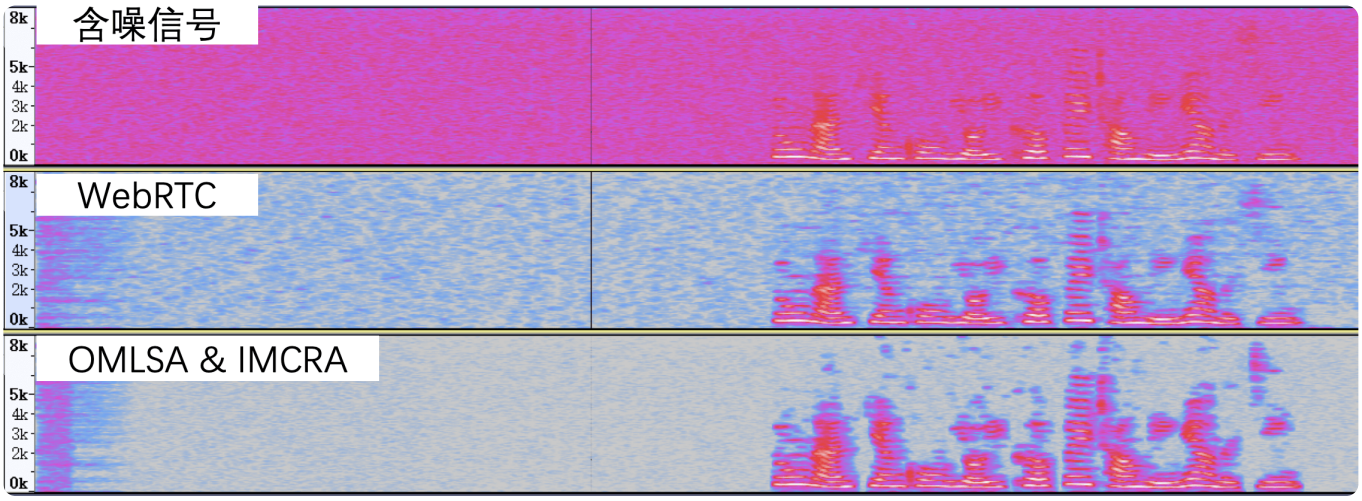

图4 WebRTC原生降噪和OMLSA&IMCRA降噪对比

从上图可以看出通过 OMLSA&IMCRA 降噪比 WebRTC 原生降噪得到的语谱更干净，残留噪声更少。

降噪算法第四招：子空间算法

子空间算法主要是针对一些已知噪声类型，量身定做一个降噪算法。其思想就是把噪声和人声投影到一个高纬度的空间，让本来不容易分离的信号变成在高纬度占据一个可分的子空间，从而可分的信号。这类算法包括非负矩阵分解和字典法建模等。

什么时候会用到这种算法呢？比如你只是要去除风噪这一种噪声，你可以用非负矩阵分解的方式单独为风噪建模，从而模型会自动消除音频中的风噪。这个在去风噪的场景下效果也是不错的。但这类方法缺点也很明显，每一种噪声都得单独建模，在噪声类型不定的情况下就很难穷尽达到好的效果。这里我们可以看一下基于非负矩阵分解的降噪。如图 5 所示：

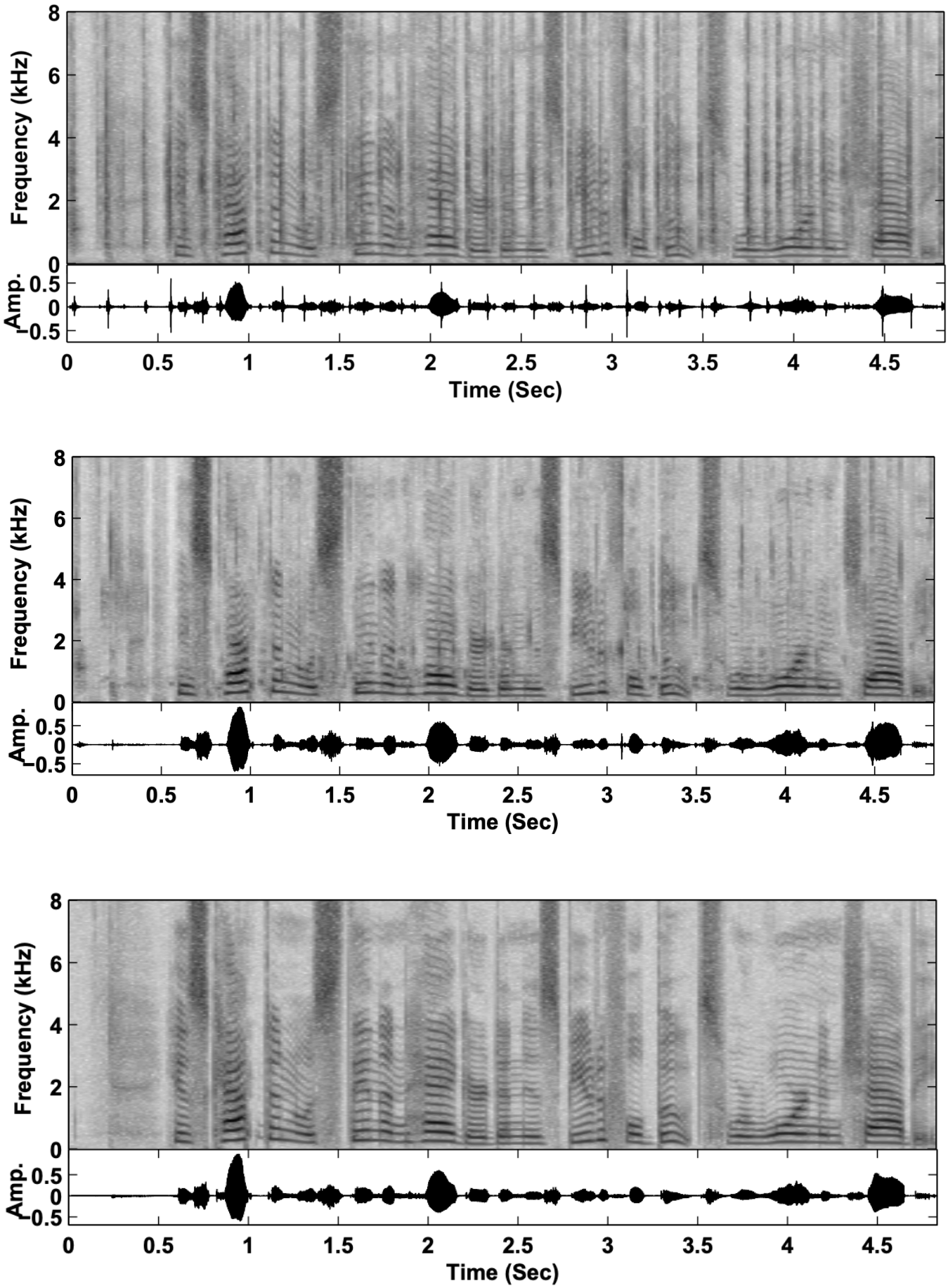

图5 非负矩阵分解消除鼠标声（从上到下依次为：含噪信号、非负矩阵分解降噪之后的信号和原始语音信号，Mohammadiha, Nasser, and Simon Dodo，2014.)

从上图可以看出通过非负矩阵分解来消除鼠标声，降噪之后的那些鼠标点击产生的黑色竖条就被消除了。

降噪算法第五招：基于机器学习的降噪

这块最近还是比较火热的。它是通过数据训练的方式，训练人工神经网络来进行降噪。特点是噪声鲁棒性好，能兼顾稳态、非稳态甚至是瞬态噪声。比如图 6 中咖啡馆的噪声属于混合类型的噪声，传统降噪算法对语谱的损伤就很大，很多高频信息都丢失了，而使用 AI 算法在保留语谱的同时又起到了比较好的降噪效果。这块我会在下节课带你详细了解一番。

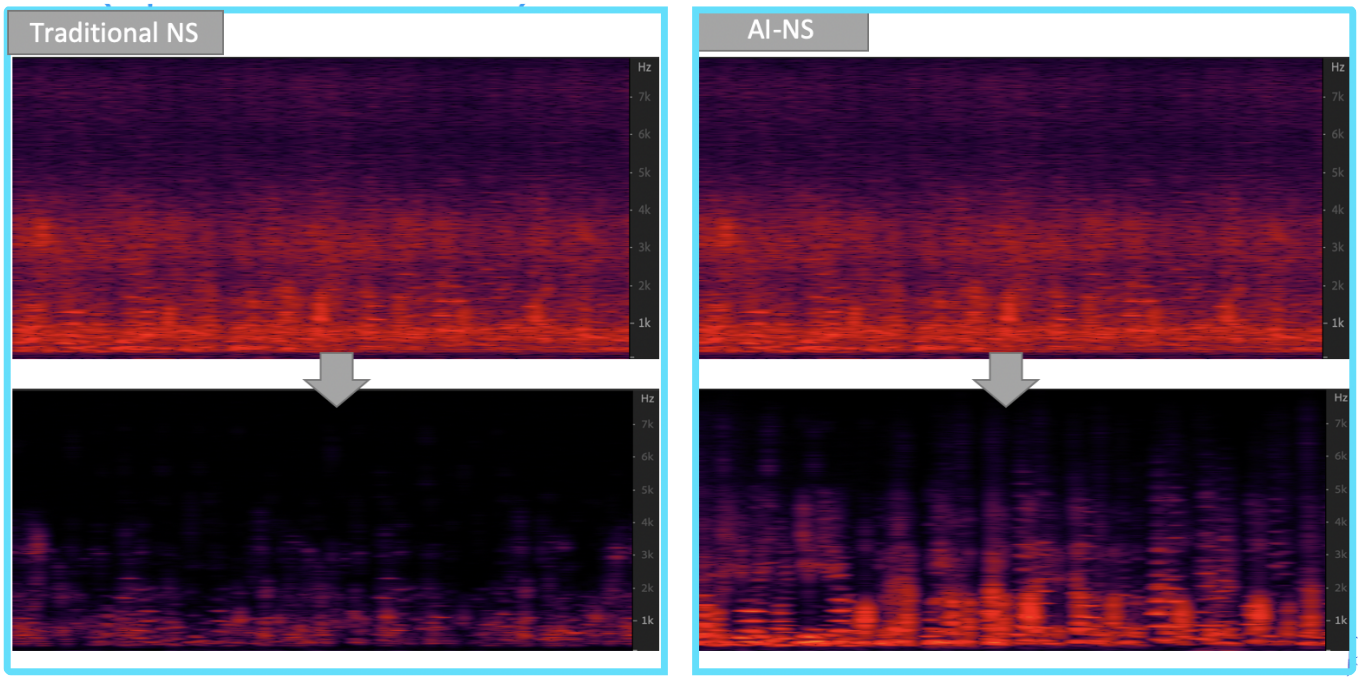

图6 传统的降噪（WebRTC）与深度学习的降噪频谱对比（咖啡馆噪声）

这里讲了很多噪声的类型以及如何消灭它们的方法。其实如果宏观地从整条音频链路的角度上来说，我们在实时音频中大部分采集的音频就是单通道了，这里介绍的算法也都是单通道降噪算法。如果采集的时候可以用多个麦克风或者麦克风阵列，则可以使用波束形成的方法先锁定声源方向来收音，比如选择说话人的方向来收音。这样采集来的信号，信噪比就比较高了，再通过单通道降噪就可以事半功倍。

小结

这里我们用图 7 来总结一下噪声分类和 5 种降噪算法及其适用范围：

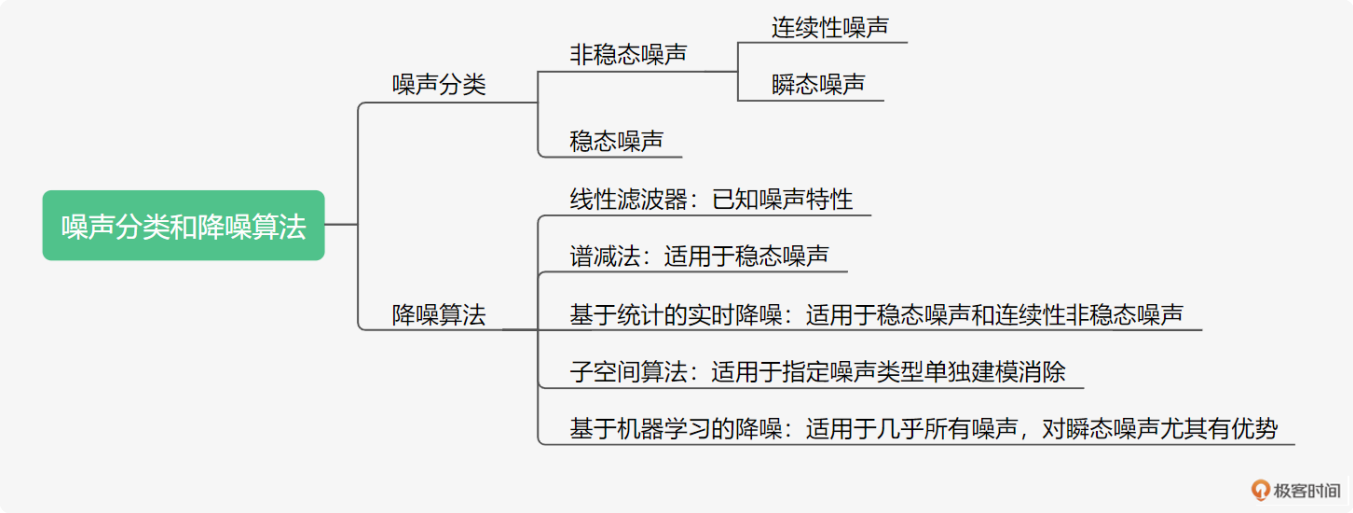

图7 课程总结

你可以针对不同的噪声类型选择合适的降噪算法。在实际使用的时候还必须结合应用场景的需要来权衡算法的复杂度来解决主要矛盾。比如，现在很多 TWS 耳机都是自带降噪算法的，但是耳机上的 DSP 芯片的算力有限，可能就用单个子空间算法来解决一下耳机常见的风噪就可以了。

还有一些音乐场景我们可能需要牺牲一些降噪性能对降噪的幅度做限制。比如，每个频点最多只能降 3dB，这样来保证音乐信号不会被削弱。所以你在实际的使用中可以从全局出发多问问自己：我应用的场景是什么？我最关心的效果是哪些？

思考题

现在请你思考一下，你在平时的远程会议里，或者微信电话里的噪声是什么类型的？在使用不同的实时互动 App 的时候，你也可以试试键盘或者鼠标的声音是否还能听得见，是不是可以根据降噪效果猜一下这些 App 里面用的是什么样的降噪算法？

你可以把你的答案和感受写下来，分享到留言区，与我一起讨论。我们下节课再见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 04 | 如何评价音频质量的好与坏？

下一篇 06 | 如何将AI技术运用到降噪中？

精选留言 (2)

写留言

Geek_7de4c5
2021-12-01

微信通话有严重的混响。并且有吃音现象。老师请问一下，对于键盘声这种瞬态噪声，webrtc原生降噪算法是否可以增加内置模板解决呢？

展开

作者回复: 其实是可以的，WebRTC中是一个对稳态噪声的降噪，可以再接一个对瞬态噪声识别加处理后处理模块。或者先识别然后修改原生降噪模块的频谱mask的结果。再或者直接用AI降噪替代。

springXu
2021-12-01

微信或者远程会议的电话里的噪音，我认为是非稳态噪音。原因稳态噪音已经被降噪了，像风的声音又或者笔记本风扇的噪音。
其他实时互动如qq视频聊天，我在一年前使用时，键盘的按键声是可以没有被过滤的。

展开

作者回复: 感知正确。